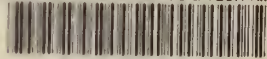


A11102 469195

REFERENCE

NBS
PUBLICATIONS

NAT'L INST OF STANDARDS & TECH R.I.C.



A11102469195

Rockett, John A./The NBS/Harvard Mark VI
QC100 .U56 NO.85-3281 1986 V19 C.1 NBS-P

-3281

The NBS/Harvard Mark VI Multi-Room Fire Simulation

John A. Rockett
Masahiro Morita

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Fire Research
Gaithersburg, MD 20899

May 1986



U.S. DEPARTMENT OF COMMERCE

NAL BUREAU OF STANDARDS

QC

100

.U56

#85-3281

1986

NBSIR 85-3281

**THE NBS/HARVARD MARK VI
MULTI-ROOM FIRE SIMULATION**

John A. Rockett
Masahiro Morita

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Fire Research
Gaithersburg, MD 20899

May 1986

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

TABLE OF CONTENTS

	<u>Page</u>
List of Figures	iv
Abstract	1
1. INTRODUCTION	1
2. THE STRUCTURE OF NBS/HARVARD MARK VI	4
2.1 The Input Module	5
2.2 The Execution Module	8
2.3 Enrichments to Harvard VI Added by NBS	11
2.4 Computer Memory Requirements, Program Transportability, Operating Speed and Convergence	13
3. CONCLUSIONS	15
4. REFERENCES	16
APPENDIX A. Comments on Mark V	18

LIST OF FIGURES

	<u>Page</u>
Figure 1. Block diagram of NBS/Harvard Mark VI	22
Figure 2. Subroutine map of input module	23
Figure 3. Subroutine map of execution module	24

THE NBS/HARVARD MARK VI MULTI-ROOM FIRE SIMULATION

John A. Rockett
Senior Scientist
and
Masahiro Morita
Visiting Scientist*
National Bureau of Standards
Center for Fire Research
Washington, DC 20234

Abstract

The NBS/Harvard Mark VI multi-room fire simulation program structure is discussed and compared with Harvard V. In addition to the current, operating version of VI, a development version is being used to test enrichments which can be readily moved into the operational version as they mature. The program is written in ANSI standard FORTRAN 77 and is transportable to computers of various manufacture.

1. INTRODUCTION

In the last decade, Professor Howard W. Emmons of Harvard University has been a leader in the development of the technology for predicting room fire behavior [1]. Supported by grants from the National Bureau of Standards, he and Dr. H. Mitler, assisted by a succession of students, produced a series of increasingly versatile computer based fire simulations - the Mark I-V simulations for a single, isolated room, and Mark VI for fires in multi-room configurations. In the following text numerous comparisons are made between the Mark V and VI computer fire simulations. For readers unfamiliar with Mark V,

*Permanent address: Department of Applied Mathematics, Science University of Tokyo, Tokyo, Japan

references [1-3], this may present a problem. For them a brief summary of the structure and operation of the Mark V simulation is presented in Appendix A.

The Mark V simulation released in 1980 [2,3] had the same basic program structure established by the work leading to Mark III, although refinements and improvements had been made. The principal advances from Mark III through V were in the amount and detail of the fire physics included. The program structure of Mark VI [4] reflects the accumulated experience of the earlier programs, but has been significantly improved relative to them. Mark VI permits fire to exist in more than one room. A consequence of this is that, potentially, a much larger number of simultaneous equations may have to be processed than was true for Mark V - for fires in n rooms, n times as many as in Mark V, plus any additional relations arising from the interactions between rooms. Thus it was important to process the mathematics very efficiently. In addition, it was hoped that some well known, if not completely understood, numerical convergence problems of Mark V could be overcome. Dr. John Ramsdell, then a Harvard computer science graduate student, was able to use the design of Mark VI as a part of his doctoral research. His work resulted in a separate computer program, "Juggle", which examines the equation set of Mark VI, for any particular fire situation, and determines an optimal solution strategy [5]. Certain features of Mark VI result, in part, from constraints imposed on it so that Juggle may be used.

Before developing the computer code for NBS/Harvard VI, several decisions were made which placed constraints of the programmer's activities. These same constraints had been applied to the development of computer code V. (1) The program should be transportable to any computer of adequate power with no

reprogramming required. This implied selection of a universally available program language for which a meaningful standard existed. FORTRAN was the only suitable candidate. BASIC and PASCAL, for example, are not well standardized; compilers for ADA were not available at the time VI was written. This also meant that high resolution graphics could not be used. Although there is a great deal of software available to produce excellent graphics using specific plot hardware, no available software is universally applicable to various combinations of the most common hardware. Some low resolution graphics has been included and more should certainly be considered for future releases of the programs. Similarly, a program modification which separated the input and execution modules (see discussion below) into two separate programs coupled via an unformatted, binary data file was discarded because the code creating and reading the unformatted file was found not to be machine independent and could not readily be made to be. (2) The program should be in the public domain. This meant that no proprietary software was used. In particular, none of the powerful, but copyrighted, numerical packs were considered. Instead, numerical procedures described in the open literature were adapted to our purpose. In some respects this was inefficient, but the resulting code could be freely distributed to interested fire scientists throughout the world. This has been very important to the acceptance of new, quantitative methods by the fire protection community. (3) The program should be usable both in interactive and batch processing modes. In contemporary computing environments, interactive program operation is essential, but, for many production uses, batch operation is highly desirable particularly with multi-tasking machines. VI (and V) can be run in batch mode but, in their present form, a preliminary interactive run is virtually required to provide a sample input file. Successive modifications of this are used to provide a

series of batch input files. (4) Finally, a high standard of internal documentation was made a significant project goal. Although internal documentation can always be made more complete and informative, VI is even more successful in achieving this goal than V. One result of the documentation effort is extensive comment headers for the three major COMMON blocks of VI. To reduce the bulk of the program listing, these were separated from their respective commons and reproduced only once (in the MAIN program) rather than at each occurrence of the COMMON.

The Harvard "Home Fire Project", one of whose outputs had been the Harvard room fire simulations, was terminated in the summer of 1983 upon the retirement of Professor Emmons. At that time Harvard VI was transferred to the Center for Fire Research, National Bureau of Standards.

Since receiving the Harvard VI computer code it has been made operational, modified to conform to the ANSI FORTRAN-77 standard and some (coding) errors in the physics corrected. Recently it has been tested against multi-room fire test data acquired in 1980 at NBS [7]. This report will discuss the Harvard VI simulation in its current form which is now designated NBS/Harvard VI. Comparison of the simulation and experimental results will be reported separately.

2. THE STRUCTURE OF NBS/HARVARD MARK VI

As stated above, the structure of NBS/Harvard VI (see figure 1) has a family resemblance to the earlier Harvard simulations and, in particular, to Harvard V. Changes between V and VI include the elimination of most COMMONS

and rewriting the physical subroutines to return only one variable each (i.e., they have the structure of function statements). In addition, a new mathematics package has been included. Two versions of the simulation are in use at NBS. One contains minimal changes from the version received from Harvard. The other is a development program with numerous additions to both the program structure and the physics incorporated. This discussion will deal primarily with the simpler version.

The simpler form of VI consists of a small main program which first calls an input module, and then an execution module. The input module consists of ten subroutines and a data block. The data block contains "default" values for all the data required to run the program for a single-room "test" case (the same test case built into Mark V). The input subroutines allow the user to alter any or all of the pre-set default input and add extra input for additional rooms and objects. The result is then displayed on the user's video display terminal, for inspection and possible change (using RECAP). Those familiar with Harvard V will find the input process much like that of the older program, but somewhat easier to use.

2.1 The Input Module

Figure 2 shows the input subroutine structure. There are two parts to the input process: first (via SELSUB) the user is asked to select the physical subroutines to be used for computing the temperature distribution within objects (TMP0) and the extinction coefficient of the gas layers in the rooms (ABSRB). In each case, several alternatives are available. The program does not provide any guidance on which of the choices to use; the user

must be familiar with the routines available from reading the documentation [1-4]. Next, a dialogue is set up between the program and the user to generate the desired input:

- * INPUT3 through successive calls to DISP (which, in turn, calls DSPASK and IASK02) displays the current values for the
 - geometry of each room in turn
 - the geometry of each object in the room
 - the thermophysical/chemical properties of the objects
 - the thermal properties of the walls surrounding the room
 - properties not specifically room-associated, such as the plume entrainment coefficient, ambient air properties, ...
- * The user refers to tabulations of the input presented on his console screen and enters any changes desired. The revised values are then displayed. This revision process can be repeated as often as necessary until the user is satisfied. Usually, all the changes can be entered on the "first pass".
- * Provision is made, via COPINP, to copy all of the values for one room for use with another room. This is useful, for example, where the building configuration is that of a motel or nursing home which contains numerous identical rooms, identically furnished.

- * Successive displays are presented for each of the subject groups mentioned above - room geometry, object geometry, etc.
- * Finally, the user is asked for the simulation time interval between (1) output summary displays on his CRT, (2) summary listings on the output (disk) file, and (3) the length of the total simulation.
- e * When all the subject groups have been reviewed, a recapitulation (in a different format) is presented for review (by RECAP). If any mistakes are detected, the user goes back to the beginning of the process, changing or accepting each input display in turn. Again the recapitulation is presented for final acceptance or yet another revision cycle.
- * A summary of the input is written, as a header, on the output file. This completes the input module's contribution.

A feature of NBS/Harvard VI is provision for saving, as a disk file, all the user's part of the input dialogue. This may be edited (with a standard text editor) to provide input files for multiple batch runs of generally similar cases -- cases where only a few input items are altered from case to case.

NBS/Harvard VI is dimensioned for five rooms. Although it would require a program change to do so, this could readily be increased. The dimensioning is done through parameters and the required changes would be localized. All the rooms must be on a single level. This restriction has been applied because the project managers decided that the physics of buoyant flow up stair

wells and vertical shafts is insufficiently understood to allow physically reasonable algorithms to be written for the various flow regimes which could arise and to define the criteria for selecting the appropriate regime for a given set of circumstances.

2.2 The Execution Module

The execution module (figure 3) consists of four groups of subroutines:

- * A control shell, DBLS, which calls NWSTAT, NEWSTP, UNPDSC and ALDIFF. DBLS also calls the output subroutine, WRIT, at prescribed simulation time intervals. NWSTAT is called after each computational discontinuity (any discrete change in the status of the calculation such as the ignition of a previously non-burning object). These discontinuities may be planned -- events occurring at a specified time -- or unplanned -- such as radiation stimulated ignition of an object. The occurrence of the latter is detected by UNPDSC. NEWSTP is called at the beginning of each new time step. It replaces values stored from the previous, successful time step with new, converged values and makes other needed time step initializations. ALDIFF controls the numerical calculation within a time step.

- * A mapping between the physical variables and the numeric package. This is called from NWSTAT and consists of UNPACK, PACK, SETSYS, and PUTPTR. A somewhat simplified statement of this is that three vectors are used to define the simulation at any time: one is the maximum set of variables available (dimensioned) to describe the fire's effects. The

second is a subset of this, the variables actually in use at any time. The third is a reordering of the second set so that the variables are conveniently grouped (packed) for use by the mathematics. Note that the equation set to be solved may change during the calculation. As discussed above, such a change is detected by UNPDSC and implemented by NWSTAT. Whenever such a change occurs the pointers which define the vectors must be re-set. This is done by SETSYS and PUTPTR.

- * An algebraic-differential equation solving package. Like Harvard V a Gauss-Seidel algorithm is used, switching to Newton's method when Gauss-Seidel fails to converge. The algorithms used by V were completely rewritten for VI however. The equation solver is implemented by ALDIFF through calls to EVJAC and CORECT. When the Gauss-Seidel method is used, CORECT, iteratively sets up a trial solution vector through calls to FUNCT and SOLVE while EVJAC calls FUNCT and DECOMP to create the matrix used by Newton's method. FUNCT calls EVALFP and ERFDE. (EVALFP and ERFDE are discussed in the next paragraph.) Unlike Harvard V, which (except when convergence problems arise) uses a user input time step, NBS/Harvard VI uses a variable length time step. The step length is related to the previous successful step. NBS/Harvard VI operates in double precision (i.e., minimum 64 bit words).

- * A set of physical subroutines. These contain the actual fire physics to be included. Note that it is the particular set of physical subroutines selected which make the otherwise quite general simulation program into a multiroom-fire simulation. There are fifty physical

subroutines in the present version of Harvard VI. At each time step they are called from EVALFP (fixed point equations) and ERFDE (physical quantities whose expression requires root finding or the solution of a differential equation).

To facilitate the development of Harvard VI, the physical content was frozen at the official Harvard V level current at the time coding of VI was started (1981). (Note that there are versions of Harvard V in use which have been variously "enriched" since 1981.) However, as noted above, two significant changes were made to the physical subroutine structure when the Harvard V routines were moved into VI. In Harvard V, subroutines were set up to deal with a "subject". Thus subroutine LAYR04 of Harvard V returned values for eleven attributes of the upper hot layer; GFIR03, five attributes of a growing fire. In order that "Juggle" be able to operate, it was necessary that each physical variable be computed in its own subroutine. This would allow the order of up-dating of the variables to be varied in accordance with the optimization determined by Juggle. Thus the first change was to divide each subject related subroutine of V into a set of subroutines, each returning only one variable. For example, LARY04 had to be divided into eleven separate subroutines. The second change relates to the way information is passed to the subroutines. In Harvard VI all the information needed by a subroutine is passed in the calling statement. The last quantity in the calling list is, conventionally, the value returned. In Harvard V each physical subroutine's calling statement contains the names of the variables returned. Other quantities needed to compute the subroutine outputs are passed via labeled COMMON. The calling structure of VI, adopted partly to accommodate Juggle, is very significant as it makes the physical subroutines independent of the rest

of the simulation architecture. Thus a library of physical subroutines can be built which will support Harvard VI, but can also support other simulations developed for other purposes. This is a sufficiently great advantage that the current NBS enriched version of V is being altered to use the VI physical subroutines. Subroutines incorporating physics not yet a part of VI are also being put in this "transportable" form.

The version of Harvard VI delivered to NBS did not include a debug capability, for example, one similar to V's subroutine DEBUG. An extensive debug package is now included in the development version of VI. This is overlaid on the VI structure. It is of great use in program development. In its present form, however, it penetrates to the physical subroutine level making necessary an alternate set of physical subroutine codes. These alternate subroutines contain not only fire physics but also debugging information. Thus they are not independent of the particular simulation with which they operate; they are not transportable to other fire simulations containing different (or no) debugging instructions.

2.3 Enrichments to Harvard VI Added by NBS

A feature added to the development version of NBS/Harvard VI is a library of material properties. The user may either specify individual material physical and chemical properties or may select a set of properties from the library by naming the material. The library currently includes twenty-two materials or assemblies (such as a "mattress"). Current work will expand this list to include typical wall, ceiling and floor covering materials. Other changes now in process make maintenance of the materials data bank independent

of the fire simulation program and make maintaining the integrity of the data bank easier in multi-user environments.

Other enrichments to the original version of VI are: (1) A pre-programmed gas burner algorithm has been added. This allows the burner gas flow rate at selected times to be specified. The gas flow rate to the burner varies linearly with time for times between these specified points. (2) A variable vent flow coefficient algorithm has been included. It is based on the work of Steckler, Baum and Quintiere [6] who have developed a theoretical vent flow coefficient expression.

The Harvard VI simulation assumes that, within each room, there are two gas layers: a lower, cool layer of air at ambient temperature and uncontaminated by smoke, and an upper layer of hot combustion products. The layers are stably stratified and the simulation assumes a sharp interface between the layers. Observation of the actual gas layers developed in fire tests suggest that this is frequently a reasonable approximation of the actual situation, but it is a idealization. In particular, the interface between the gas layers is not sharp; in some cases there is a relatively narrow transition region across which the gas temperature and the smoke opacity changes rapidly. In others the transition region may be quite broad. Further, in many situations, the gas in the lower part of the room is neither free of smoke nor at ambient temperature [8]. Coding to account for the effect of mixing across the interface is not included in VI. Mixing occurs on both sides of a vent. Cool air flowing into a room through the lower part of a vent entrains hot gas flowing out the same vent, this hot gas is mixed with the cool air and returns some of the hot gas to the room thus raising the temperature of the cool layer and

polluting it with smoke and combustion products. On the other side of the vent, hot gas flowing out of one room into an adjacent one may entrain cool gas from the lower layer of the adjacent room. This cools the hot layer gas and adds oxygen to it, possibly allowing combustion to occur in the adjacent room if excess fuel is in the hot gas leaving the original room. Enrichments to Harvard V account for the first of these - mixing of hot gas with cool air entering the lower part of a vent [9]. This enrichment produced noticeably more realistic simulations with V. Since it is a single room model, no adjacent room is considered and the second type of mixing would not arise. Both types of mixing should be included in future versions of VI.

Other enrichments to NBS/Harvard VI are being developed but have not progressed far enough to be discussed yet. Some, like vent mixing just discussed, are already in enriched versions of V, others have not yet appeared in any room fire simulation.

2.4 Computer Memory Requirements, Program Transportability, Operating Speed and Convergence

Because the input and execution modules operate separately and sequentially, the execution module can be overlayed on the memory occupied by the input module to reduce the total memory requirement of the program by 23.59 KB out of a total of 407.5 KB.* In terms of memory requirements, overlaying the input routines is not a major advantage.

*With Perkin Elmer FORTRAN VII 0, the code unique to Harvard VI occupies 110.86 KB, COMMONs 25.2 KB and local data 196.44 KB. 75 KB are needed by the FORTRAN library. Code for the input routines uses 14.43 KB and local data 8.64 KB. One COMMON used only in the input section requires 0.52 KB.

Operation of Harvard VI depends, of course, on the particular computer used. The version of Harvard VI described above was developed and runs on a VAX 11-780. Since bringing it to NBS it has been run on a UNIVAC 1100-82, an IBM 4341 (IBM vs FORTRAN, level 1.3 FORTRAN verifier used to check the Harvard VI source against the ANSI FORTRAN 77 standard), a Perkin Elmer (P-E) 3242, a CDC 855 and an IBM XT (equipped with a 8087 math co-processor). Execution speed depends on the particular case (input data), machine and FORTRAN compiler used. The following execution times are presented as an indication of execution speed, they should not be considered as precise numbers or to indicate execution times to be expected by other users with other machines. The times are for the P-E 3242 using the P-E optimizing compiler (FORTRAN VII 0 release R05-01.00).

- * Standard test case (built-in, default values for all input items) of one room, two objects, the one initially ignited is a "growing fire": calculation up to 500.0 simulation seconds, CPU time 768.224 seconds. Harvard V requires 792.64 CPU seconds for the same single room test case and a similar numerical convergence condition.
- * Five room simulation of the NBS multi-room fire tests using a single gas burner fire (to be discussed in a paper to appear separately): calculation up to 500.0 simulation seconds, CPU time 1034.659 seconds.

Very few convergence problems have been encountered to date with NBS/Harvard VI. Note, however, that we have not run a large variety of input in the brief time the program has been in use. Computational times

occasionally become excessive for particular input combinations, especially for very small vents (near the floor) combined with large pyrolysis rates per unit room volume. In one such case it was found that the solution values were oscillating, with small amplitude, around slowly changing values. By changing the minimum time step from 0.0001 to 0.001, computing time was reduced in the ratio of 56 to 1 with no significant change in the solution. While this "fix" worked for this case, it is not a general fix. In many cases it has been found that a tight convergence criterion may be advantageous while, for a few other problem input data sets, relaxing it helps. The default convergence criterion can often be relaxed with very little change in the computed results while yielding a significant reduction in computing time. Thus a two order of magnitude loosening of the convergence criterion for the standard test case cited above reduced computing time by a factor of four with no significant change in the results. The default value was made relatively stringent to reduce the number of cases which encountered numerical difficulties while retaining acceptable computing times.

3. CONCLUSIONS

NBS/Harvard VI is a significant advance over Harvard V both because of its multi-room capability, and because of improvements in the details of its program structure. The physical subroutine structure of VI is a major advance through its decoupling of the physical phenomena from the simulation's program structure. This not only facilitates building a simulation independent subroutine library, but isolates better the individual physical processes making easier incremental improvement of the physics.

The input structure has also been streamlined and improved. The use of a materials data bank, now in the development version, will be a further significant improvement as many users have found it very difficult to locate suitable data from the current fire literature for use as input for the Harvard series of fire simulations.

Although the computational process used in VI is similar to that of V, the way in which it is built into the program will facilitate introduction of another "math pack" should one seem desirable.

4. REFERENCES

- [1] Emmons, H.W., "The prediction of fires in buildings", Seventeenth Symposium (International) on Combustion, pp. 1101-1111, The Combustion Institute, Pittsburgh, PA (1978).
- [2] Mitler, H., "The Physical Basis for the Harvard Computer Fire Code", Home Fire Project Technical Report No. 34, Division of Applied Sciences, Harvard University, Cambridge, MA, October 1978.
- [3] Mitler, H.E. and Emmons, H.W., "Documentation for CFC V, The Fifth Harvard Computer Fire Code", Home Fire Project Technical Report No. 45, Division of Applied Sciences, Harvard University, October 1981.
- [4] Gahm, J.B., "Computer Fire Code VI", NBS-GCR-83-451 (2 volumes), U.S. Department of Commerce, National Bureau of Standards, Center for Fire Research, Gaithersburg, MD 20899, December 1983.
- [5] Ramsdell, J., "Variable Elimination in the Home Fire Code", ASME/AICHE National Heat Transfer Conference, Milwaukee, WI, August 2-5, 1981.
- [6] Steckler, K.D., Baum, H.R. and Quintiere, J.G., "Fire Induced Flow Through Room Openings - Flow Coefficient", NBSIR 83-2801, U.S. Nat. Bur. of Stds., Gaithersburg, MD 20899, March 1984. See also 20th Symposium on Combustion (International), The Combustion Institute, Pittsburgh, PA (1985).
- [7] Cooper, L.Y., Harkleroad, M., Quintiere, J.G. and Rinkinen, W., "An experimental study of upper hot layer stratification in full-scale multiroom fire scenarios", Jour. of Heat Transfer, Vol. 104/741, November 1982.

- [8] McCaffrey, B.J. and Rockett, J.A., "Static pressure measurements of enclosure fires", Jour. of Research of the National Bureau of Standards, Vol. 82, No. 2, Sept-Oct 1977, pp. 107-117.
- [9] Rockett, J.A., "Modeling of NBS mattress tests with the Harvard Mark V fire simulation", Fire and Materials, Vol. 6, No. 2, 1982, pp. 80-95.

APPENDIX A. Comments on Mark V

A computer simulation of any physical process must provide five functions:

- (1) An input module through which the user enters the data necessary to fully specify the particular problem to be solved.
- (2) An output module through which the computer presents the results of its calculations to the user.
- (3) A mathematics module which sets and solves (numerically) the equations of the simulation.
- (4) A set of subroutines which define the actual physical process to be simulated. The amount and detail of relevant physics accessible to the simulation is determined by the content of the physical process module.
- (5) A coordinating shell module which controls and modulates the entire computation.

The input module of Mark V is comprised of eight subroutines. Mark VI is, as mentioned in the main text, functionally very much the same. Evolution of the input module (Marks III-V) has been toward making the program more "user friendly", though much more could be done. When using V in interactive mode, the user is presented with a succession of "screens" displaying the

current values of a related set of input items (such as physical/chemical properties of an object). These items are numbered. To alter any item on the screen, its number is entered. When all the numbers of items to be changed have been entered, the user is prompted to enter new values for the specified items. The revised screen is re-presented for further revision of the user advances to the next screen.

One of the most difficult features in using the Harvard codes is finding suitable input data for the fire properties of materials. This reflects the rich physical content of the simulations: many effects are modeled, but with each added effect, additional input is required. Users of V find a great deal of data asked for, much of in a form, or units, they are not accustomed to.

The output modules of V and VI are also very similar. The computational results are presented as a table: variable name and value, printed at regular intervals. One weakness of this is the very complexity of the simulation. Much is taken into account and much is reported. With the inclusion of more physics the situation can only get worse. An alternative would be to provide some form of graphical output but, as mentioned in the main text, this is currently technically impossible if the program is to be transportable between a wide variety of machines. A number of users of V have developed their own plot interface. Typically this is an offline program which reads the Harvard output and reformats it in a form suitable for input to that particular user's plot hardware/software. Some changes in the present output could facilitate this translation, but a local (user supplied) interface would still be needed.

The early Harvard codes, Mark II-IV, saw the development of the present Mark V math pack. The equations solved are non-linear, coupled algebraic and differential equations and require an iterative solution scheme. The Mark V math pack has been tailored to the particular demands of its equation set. It uses a modified Gauss-Seidel successive substitution algorithm with a (user alterable at input) fixed length time step. If this fails to converge in a (input alterable) number of iterations, the time step is halved and the failed step re-attempted. The next few time steps proceed at the most recent, successful time step. When repeated successes have been found, the time step is doubled. With continued success, this doubling is repeated until the original (maximum) time step is achieved. If, on the other hand, time step halving proceeds until a (built-in) minimum step is unsuccessful, an alternative, Newton's method is adopted. Convergence is based on all the variables currently active in the system regardless of their type: algebraic, differential or integral. Over the years the math structure of V has undergone considerable optimization and appears to be both efficient and robust. Nevertheless, V encounters convergence problems with some input data sets. Undoubtedly some of these problems relate to the imperfect modeling of the fire physics, but some may arise within the math pack. The adoption of an alternative set of math algorithms in VI was an attempt to reduce the number of these problem data sets.

As mentioned in the main text, the physical subroutine content is the same in V and VI although the subroutine architecture has been improved in VI. This is immaterial to the user but affects program development.

The coordinating modules of V and VI are more similar in concept than an initial study of the computer codes would suggest. One feature of both is a mapping between the variable set as treated by the math module and that treated by the physical process subroutines. The Mark VI version of this is discussed in the main text; it is noticeably simpler in V. The user has no contact with this mapping process, but it must be adjusted when certain types of program changes are made (for example, new physics added). The shell structure of V also includes a debugging capability. The debugging process may be initiated at any (pre-selected) time in the simulation. Various features are provided of which three of the more useful are: (1) a message giving the name of each physical subroutine upon exiting from the subroutine, (2) presentation of a table of the names and values of all variables in the system at the completion of each iteration, and (3) the name and normalized fluctuation of the variable with the largest fluctuation at the end of an unsuccessful iteration process.

HRVD6

Main program: calls BINP (which assembles the input data)
and DBLES (which carries out the computation)

-BINP

Select alternative physical subroutines
Modify input from default to desired case
Specify frequency of output and total run length
Summarize input as header on output file

-DBLES

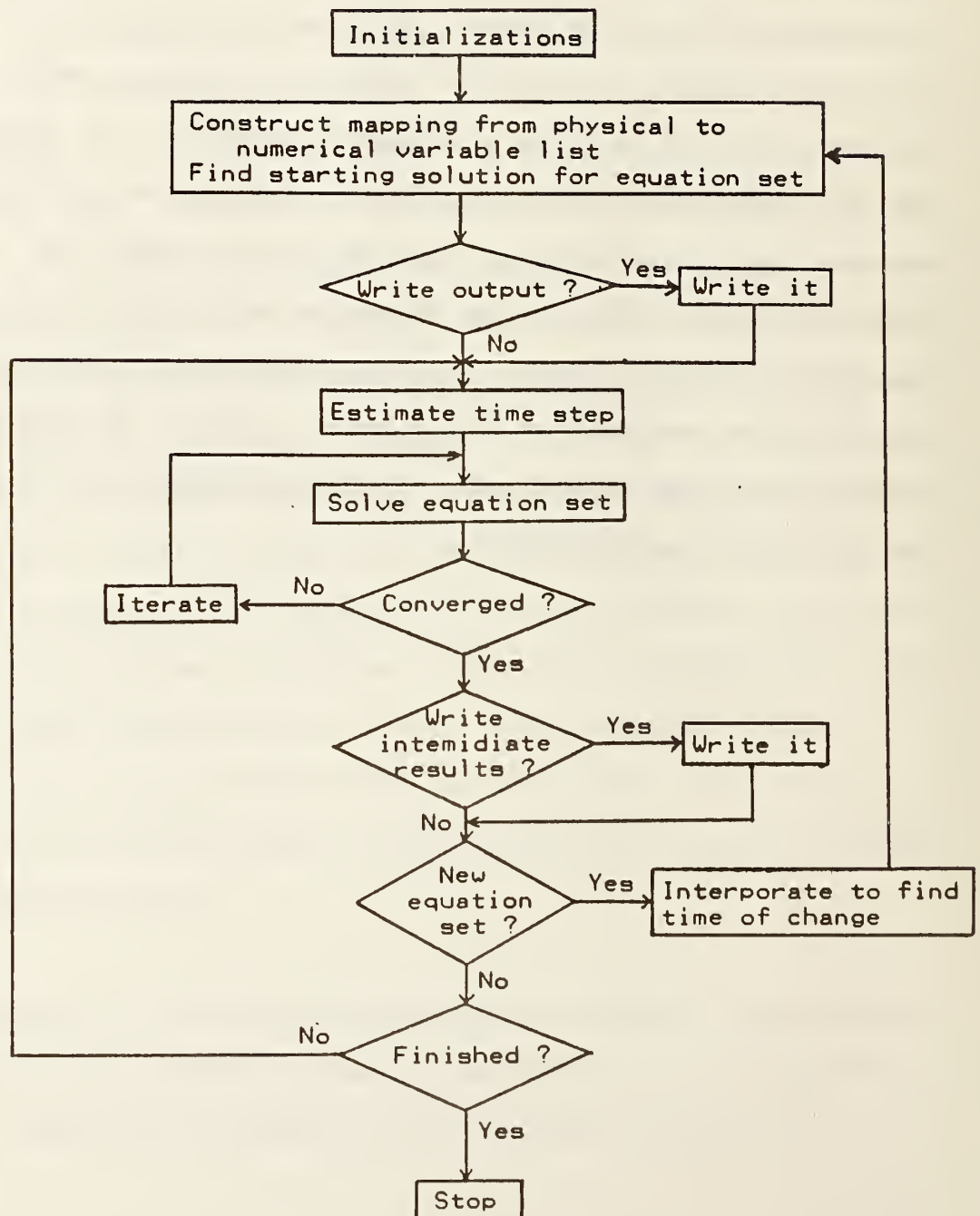


Figure 1. The structure of NBS/Harvard VI.

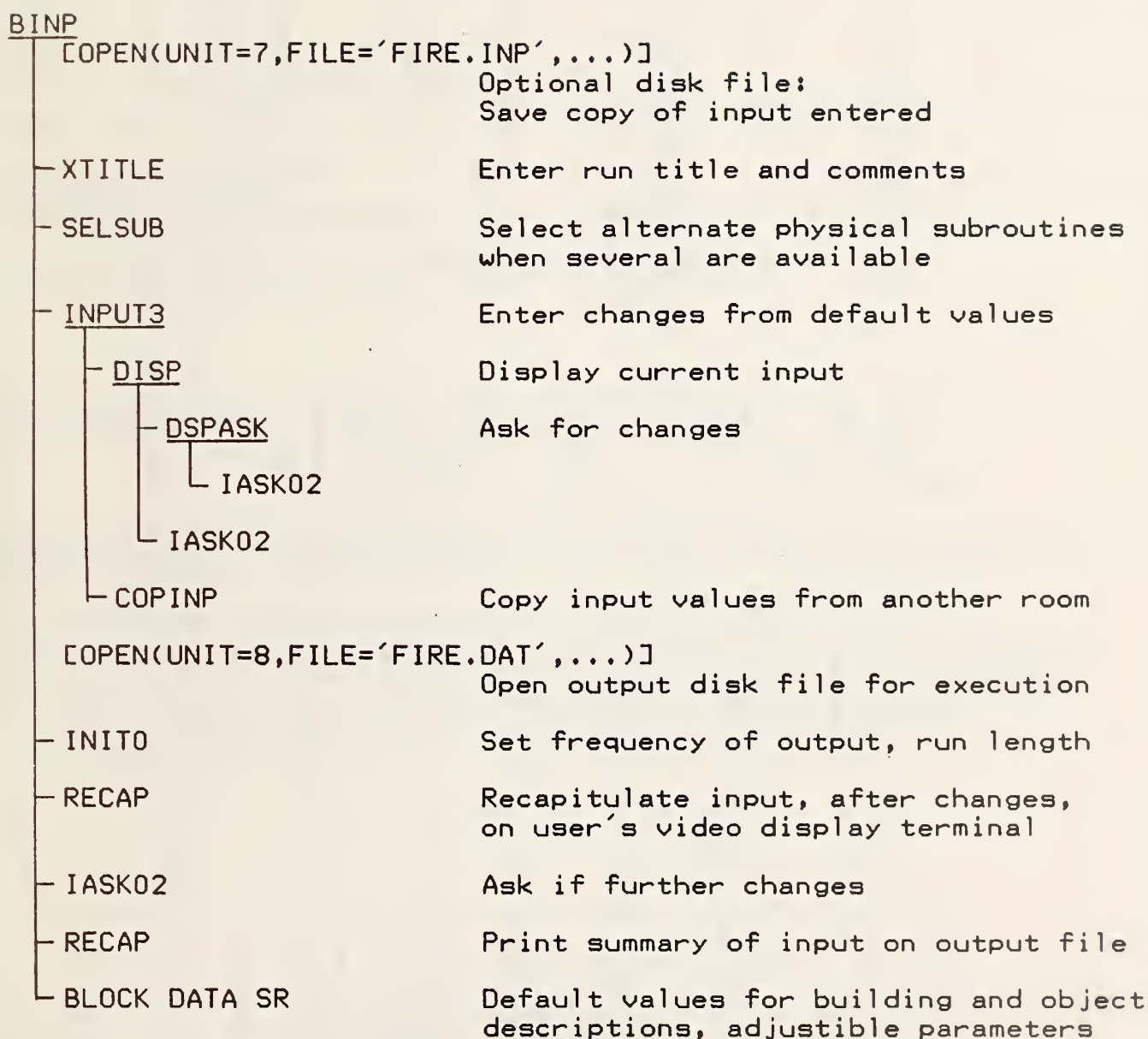


Figure 2. Subroutine map of BINP, the input module of NBS/Harvard VI.

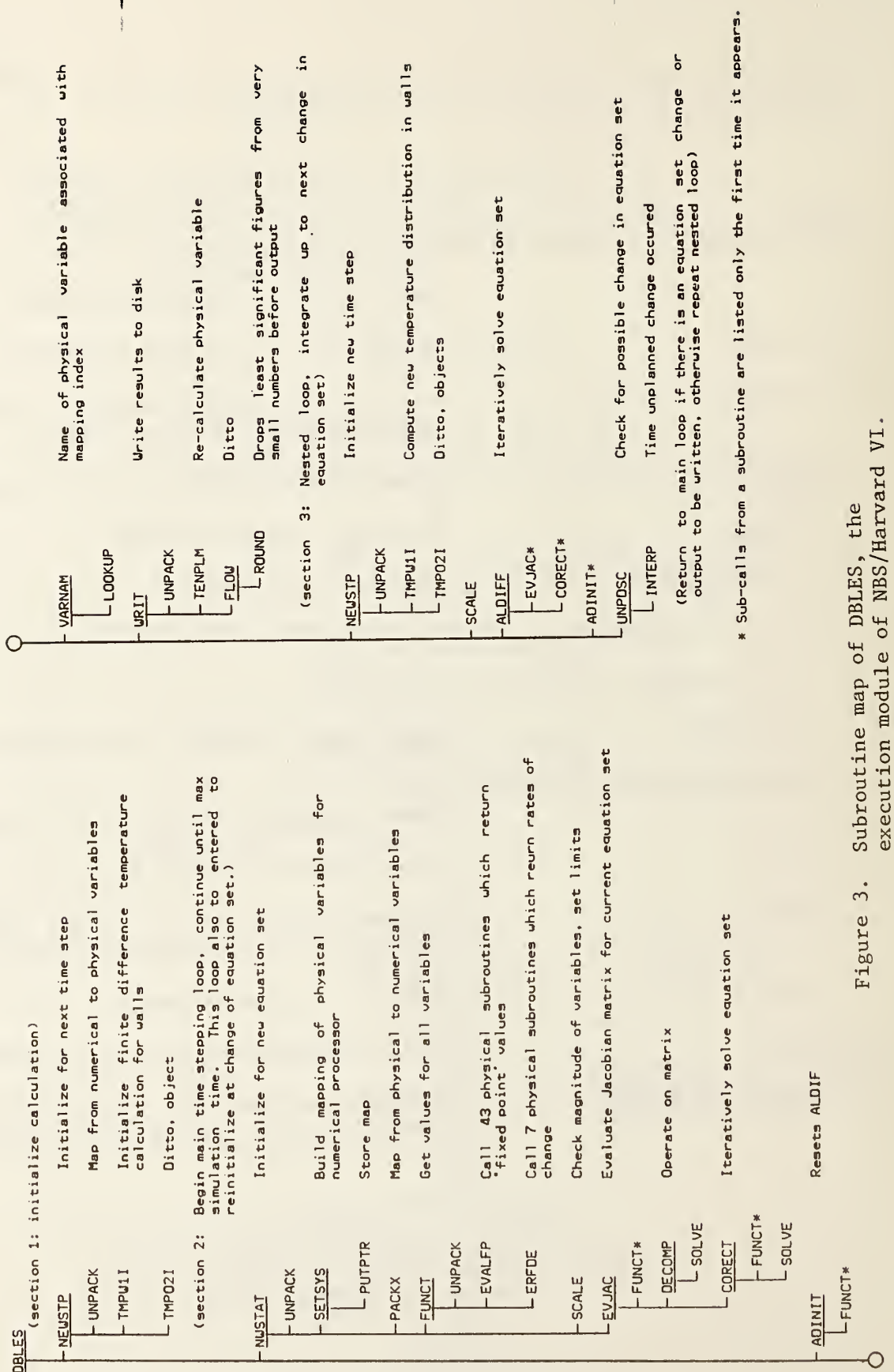


Figure 3. Subroutine map of DBLES, the execution module of NBS/Harvard VI.

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)		1. PUBLICATION OR REPORT NO. NBSIR-85/3281	2. Performing Organ. Report No.	3. Publication Date May 1986
4. TITLE AND SUBTITLE The NBS Harvard Mark VI Multi-Room Fire Simulation				
5. AUTHOR(S) John A. Rockett and Masahiro Morita				
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE XXXXXXXXXXXX Gaithersburg, Maryland 20899			7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)				
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) The NBS/Harvard Mark VI multi-room fire simulation program structure is discussed and compared with Harvard V. In addition to the current, operating version of Mark VI, a development version is being used to test enrichments which can be readily moved into the operational version as they mature. The program is conservatively coded and readily transportable to computers of various manufacture.				
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) computer models; compartment fires; fire models; multiroom fires; simulation				
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 29 15. Price \$9.95	

